Claims

1.   In a software system including a standard mechanism for accessing properties, the standard mechanism including:

a first operation for obtaining a property identifier;

5   a second operation for obtaining a property value; and

a third operation for setting the property value,

an object comprising:

a property, the property comprising a property identifier and a property value;

an implementation of the first operation;

10   an implementation of the second operation; and

an implementation of the third operation, the implementation of the third operation setting both the property identifier and the property value if the third operation is executed for a first time, and changing the property value to a specified new property value if the third operation was previously executed.

15

2.   The object according to claim 1 wherein the property further comprises a property type and wherein the implementation of the third operation sets the property type in the first entry if the value for the first property has been previously set, the implementation of the third operation sets the property type in the first entry in the table if the value for the first property has not been

20   set.

3.   In a software system including a standard mechanism for accessing properties of objects, the standard mechanism including:

a first operation for enumerating property identifiers;

25   a second operation for obtaining a property value of a property identified by a property identifier; and

a third operation for setting the property value of a property identified by a property identifier,

an object comprising:

a table containing a plurality of entries, each entry comprising a property identifier and a property value;

an implementation of the first operation, the implementation of the first operation retrieving a first property identifier of a first property from one of the entries in the table;

5   an implementation of the second operation, the implementation of the second operation obtaining the property value from the one entry;

an implementation of the third operation, the implementation of the third operation setting a property value in the one entry if a value for the first property has been previously set, the implementation of the third operation setting a property identifier and a property

10   value in the one entry in the table if a value for the first property has not been set.

4.   The object according to claim 3 wherein the each entry further comprises a property type and wherein the implementation of the third operation sets the property type in the first entry if

15   the value for the first property has been previously set, the implementation of the third operation sets the property type in the first entry in the table if the value for the first property has not been set.

5.   The object in claim 3 further comprising a terminal through which properties are

20   accessed and their values from the first table.

6.   A copier object in a software system, the copier object comprising:

a first terminal through which the copier object requests enumeration of property identifiers;

a second terminal through which the copier object requests obtaining property values;

25   a third terminal through which the copier object requests setting property values;

a fourth terminal through which the copier object request receipt of a trigger signal, and upon receipt of the trigger signal the copier object obtains a first property name identifier through the first terminal, through which the copier object requests obtaining a first property value using the first property identifier through the second terminal, and through which the copier object

requests setting the first property value using the first property identifier through the third terminal.

7.   A system of objects in a software system having a data memory, the system comprising:

an extractor object for extracting first encoded values from the data memory and storing them in the data memory in native machine format;

a stamper object for storing second encoded values into the data memory, the second encoded values obtained from the data memory in native machine format.

8.   The system in claim 7 the system wherein the data memory is an event object.

9.   A system of objects in a software system, the system comprising:

a container object for storing a plurality of data values;

an extractor object for extracting encoded data from data memory and storing the encoded data in the container object;

a stamper object for obtaining the plurality of data values from the container object and storing them as encoded data in the data memory.

10.   The system in claim 9 further comprising a comparator object for comparing a first data value of encoded data from the data memory to a second data value from the container object and sending a reference to the data memory to a first terminal if the first value is less than the second value, to a second terminal if the first value is equal to the second value, and to a third terminal if the first value is greater than the second value.

11.   The system in claim 9 wherein the data memory is an event object.

12.   The system in claim 10 wherein the data memory is an event object.

13. The system in claim 9 further comprising an arithmetic-logic-unit object for performing arithmetic operations on data values in the container object.

14. A method in a composition-based software system for transferring data values in event objects, the method comprising the steps of:

extracting a first value from a first event object;

storing the first value into a container object;

loading the first value from the container object;

storing the first value into a second event object.

15. The method of claim 14 further comprising the step of modifying the first value in the container object.

16. The method of claim 14 wherein the first event object and the second event object are the same event object.

17. A method in a composition-based software system for manipulating encoded data values in event objects, the method comprising the steps of:

extracting a first value from a first data field of a first event object;

decoding the first value into a normalized form;

storing the first value into a second data field of the first event object;

performing an operation that modifies the first value in the second data field, resulting in a second value being stored in the second data field;

loading the second value from the second data field;

storing the second value into the first data field.

18. A system of interconnected objects in a software system, the system comprising:

an extractor object for extracting a first value from a first data field in a first event object and storing it into a second data field in the first event object;

a modifier object for modifying the second data field;

a stamper object for loading a second value from the second data field and storing it into a third

5   data field in the first event object.

19.   An object in a software system, the object comprising:

a first terminal through which the object receives a source event;

a first offset property specifying starting offset in the source event;

10   a size property specifying size in the source event;

a second offset property specifying starting offset for merging;

a reference to a data memory for storing a data portion from the source event, starting from offset specified by the offset property and of size specified by the size property;

a second terminal through which the object receives a merge event;

15   a third terminal through which the object sends the merge event, the merge event modified by storing the data portion into the merge event at offset specified by the second offset property.

20.   The object in claim 19 wherein the first terminal and the second terminal are the same terminal.

20

21.   An object in a software system, the object comprising:

an input terminal through which the object receives an input event;

a first output terminal through which the object sends an event containing a first portion of the input event;

25   a second output terminal through which the object sends an event containing a second portion of the input event;

a first property specifying the size of the first portion.

22.   An object in a software system, the object comprising:

a first input terminal through which the object receives a latch event;

a second input terminal through which the object receives a trigger event;

a field for storing a reference to the latch event when received on the first input terminal;

an output terminal through which the object sends the latch event when the trigger event is

5   received through the second input terminal.

23.   An object in a software system, the object comprising:

an input terminal through which the object receives a first input signal;

an output terminal through which the object sends the first input signal;

10   a factory terminal through which the object requests the creation a new object instance when the

object receives the first input signal;

a property terminal through which the object requests the setting of properties on the new object

instance.

15   24.   The object in claim 23 further comprising a parameterization terminal through which the

object sends a parameterization signal so that an external object can parameterize the new object

instance.

25.   A system of interconnected objects in a software system, the system of interconnected

20   objects comprising:

a factory object for receiving creation and destruction events;

a dynamic container object for containing objects created by the factory object.

26.   An object in a software system, the object comprising:

25   an input terminal through which the object receives events;

a property specifying a target number of events;

a field for maintaining a count of events received through the input terminal;

a first output terminal through which the object sends events received through the input terminal

when the count of events reaches the target number.

551

27.   The object in claim 26 further comprising a reset terminal through which the object receives a request to reset the count to zero.

5   28.   The object in claim 26 further comprising a second output terminal through which the object sends events received through the input terminal when the count of events is under the target number.